

Objektorienterad programmering

Föreläsning 15

© Copyright
Mahmud Al Hakim
mahmud@dynamicos.se
www.webacademy.se

Agenda

- Grafiska användargränssnitt
(*GUI Graphical User Interface*)
 - Standardklassen (Control)
System.Windows.Forms.Control
 - Att dölja kommandofönstret!
 - Klassen Label
 - Att förankra och docka komponenter
 - Händelser
 - Klassen Button

Grafiska användargränssnitt

- För att det skall vara enkelt att konstruera grafiska program finns det i C# ett antal standardklasser vilka man kan använda för att enkelt skapa grafiska komponenter som kan ingå i programmen.
- Dessa klasser är indelade i två huvudkategorier
 1. Fönsterkomponenter (Controls)
 2. Verktøyskomponenter (Toolstrips)

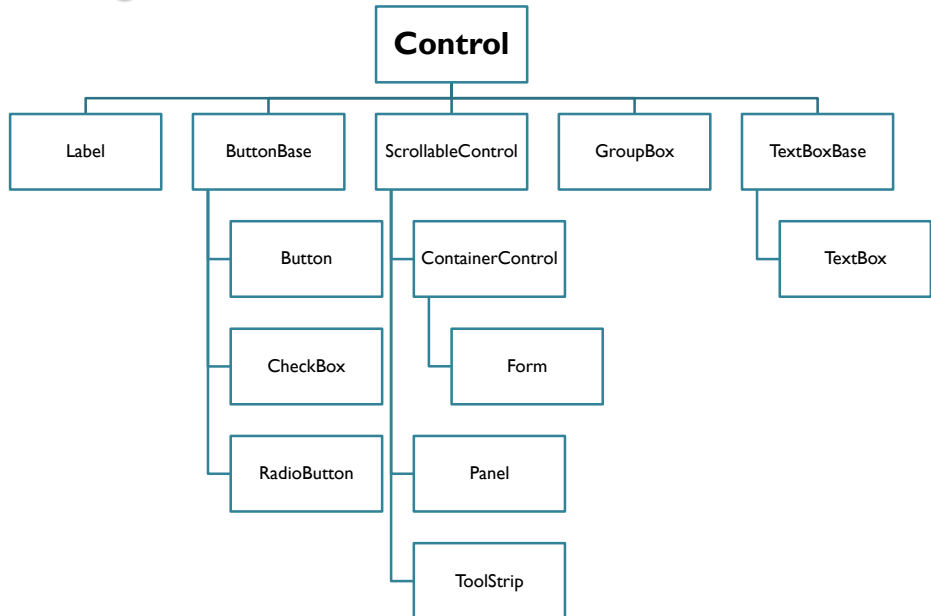
Copyright 2015 - Mahmud Al Hakim www.webbacademy.se

Fönsterkomponenter (Controls)

- Fönsterkomponenter är klasser som beskriver grafiska komponenter som antingen är självständiga fönster på skärmen (t.ex. klassen Form) eller som kan placeras direkt i sådana fönster (t.ex. klassen Label).
- Fönsterkomponenter kallas controls eftersom dessa är subclasser till superklassen **Control**.
- Klassen Control innehåller generella egenskaper som t.ex. storlek och färger.

Copyright 2015 - Mahmud Al Hakim www.webbacademy.se

Några subklasser till Control



Klassen Control

Några egenskaper

- **Name** komponentens namn
- **Text** den text som är kopplad till komponenten
- **Font** typsnitt
- **Size** storlek (bredd x höjd)
- **MinimumSize** minimal storlek
- **Location** komponentens startpunkt (left x top)
- **BackColor** bakgrundsfärg
- **AutoSize** true om komponenten automatiskt skall anpassa sin storlek (lagom storlek)

Copyright 2015 - Mahmud Al Hakim www.webbacademy.se

Klassen Form - Exempel

```
using System.Drawing; // Här finns klassen Size
using System.Windows.Forms; // Här finns klassen Form
namespace Demo
{
    class Fönster : Form
    {
        // En konstruktor
        public Fönster()
        {
            // Ändra texten som visas i namnlist
            this.Text = "Fönster";

            // Bestäm standard storlek vid körning
            this.Size = new Size(500,500);

            // Bestäm minimal storlek vid ändring
            this.MinimumSize = new Size(300, 400);

            // Ändra bakgrundsfärgen
            this.BackColor = Color.AliceBlue;
        }
    }
}
```

Klassen Fönster ärver från standardklassen Form

Klassen Size har en konstruktor som tar emot två parametrar bredd och höjd storleken anges i pixlar

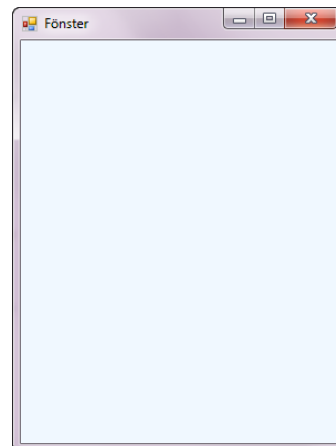
Color är en struct som finns i Drawing

Copyright 2015 - Mahmud Al Hakim www.webbacademy.se

Starta programmet

```
using System.Windows.Forms;

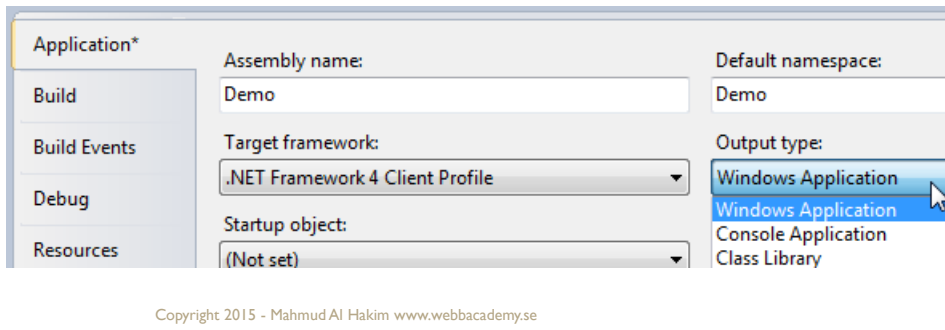
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            Fönster f = new Fönster();
            Application.Run(f);
        }
    }
}
```



Copyright 2015 - Mahmud Al Hakim www.webbacademy.se

Att dölja kommandofönstret!

- För att dölja kommandofönstret när du kör grafiska komponenter
 1. Gå till projektets egenskaper
 2. Fliken Application
 3. Välj "Windows Application" från Output type



Klassen Label

- En label (etikett) kan visa antingen en text eller en bild eller både och.
- Några egenskaper:
 - **TextAlign** placering av text
 - **Image:** refererar till en bild som visas
 - **ImageAlign** placering av bilden i etiketten
 - **BorderStyle** ramens utseende

Copyright 2015 - Mahmud Al Hakim www.webbacademy.se

Skapa objekt av klassen Label

```
namespace Demo
{
    class Fönster : Form
    {
        Label etikett1 = new Label();
        Label etikett2 = new Label();

        // En konstuktor
        public Fönster(...)
    }
}
```

Två instansvariabler
av typen Label finns
nu i klassen Fönster

Copyright 2015 - Mahmud Al Hakim www.webbacademy.se

Label – Exempel

```
// Ändra etikettens egenskaper

etikett1.Size = new Size(300, 100); // Bredd x höjd

// Bakgrundsfärg och kantlinjer
etikett1.BackColor = Color.Yellow;
etikett1.BorderStyle = BorderStyle.FixedSingle;

// Text och teckensnitt
etikett1.Text = "Hello Window";
etikett1.Font = new Font("Arial", 18, FontStyle.Bold);
etikett1.TextAlign = ContentAlignment.MiddleCenter;

// Placera etiketten i listan Controls
Controls.Add(etikett1);
```

Detta skrivs t.ex. i
konstruktorn

Viktigt
Här lägger vi
etiketten i fönstret!

Copyright 2015 - Mahmud Al Hakim www.webbacademy.se

Att lägga till barnkomponenter

- Man kan utnyttja en fönsterkomponent som behållare (containter) och lägga till andra s.k. barnkomponenter i den.
- Barnkomponenter placeras i någon av subclasserna till ScrollableControl t.ex. **Form**.
- Varje fönsterkomponent har en **lista** med namnet **Controls** vilken innehåller referenser till komponentens barnkomponenter.
- Om vi t.ex. har ett objekt, *Etikett*, som är av klassen Label så kan vi placera objektet i behållaren med hjälp av metoden Add()
Controls.Add(etikett);

Copyright 2015 - Mahmud Al Hakim www.webbacademy.se

Placera en bild i en Label

```
// Placera en bild i etikett2
etikett2.Image = Image.FromFile("blommor.jpg");

etikett2.Size = new Size(300, 300); // Bredd x höjd
// Tips! Lika stor som bilden
// etikett2.Size = etikett2.Image.Size;

// Location (position)
etikett2.Location = new Point(0,100);

// Placera etiketten i listan Controls
Controls.Add(etikett2);
```

Point är en struct som finns i Drawing och har en konstruktor som tar emot två parametrar avstånd från vänster och avstånd från topp

Copyright 2015 - Mahmud Al Hakim www.webbacademy.se

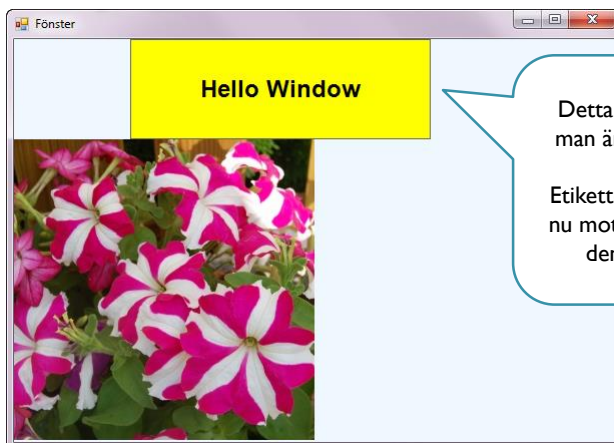
Att förankra komponenter

- Om man ändrar storleken på fönstret så behåller komponenterna avståndet till den övre och den vänstra sidan.
- Komponenterna är förankrade mot dessa sidor.
- Egenskapen Anchor bestämmer hur en komponent skall vara förankrad mot föräldrarkomponenten.
- Man kan förankra en komponent mot en eller flera sidor.

Copyright 2015 - Mahmud Al Hakim www.webbacademy.se

Anchor – Exempel

```
etikett1.Anchor = AnchorStyles.Right | AnchorStyles.Top;
```



Copyright 2015 - Mahmud Al Hakim www.webbacademy.se

Att docka komponenter

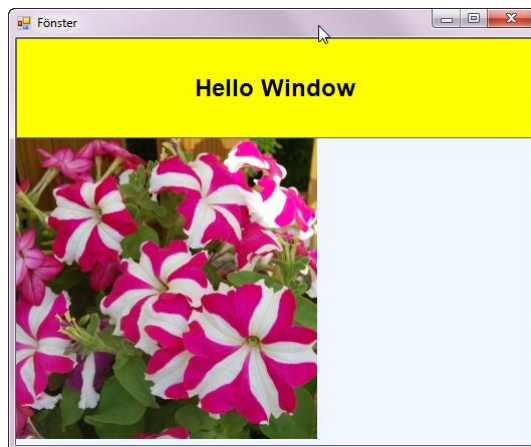
- Ett alternativt till att förankra en komponent är att använda egenskapen **Dock** för att ”docka” komponenten.
- Man anger en sida som komponenten skall läggas intill.
- Komponenterna kommer då att automatiskt anpassa sin storlek så att den fyller ut hela denna sida.
- OBS! Man kan inte kombinera egenskaperna Anchor och Dock. De utesluter varandra.

Copyright 2015 - Mahmud Al Hakim www.webbacademy.se

Dock – Exempel

```
etikett1.Dock = DockStyle.Top;
```

Tips!
Testa värdet
DockStyle.Fill



Copyright 2015 - Mahmud Al Hakim www.webbacademy.se

Händelsestyrda program

- Ett program med grafiskt användargränssnitt är händelsestyrt!
 - Exekveringen av ett händelsestyrt program sker i två faser!
 1. **Initieringsfasen**

”Tråden” som kör Main deklarerar och initierar de olika grafiska komponenter som skall användas av programmet. Här registreras s.k. **händelsehanterare** (event handlers).
 2. **Väntefasen**

Programmet ligger och väntar på att **händelser** skall inträffa. T.ex. klicka med musen eller trycka på någon knapp. När en händelse inträffar anropas den händelsehanterare som har registrerats för att ta hand om händelsen!

Copyright 2015 - Mahmud Al Hakim www.webbacademy.se

Händelsehanterare

En händelsehanterare är **en metod** med två parametra

En referens till den komponent i vilken händelsen uppstod

Referens till ett objekt till klassen EventArgs som ibland innehåller mer info om händelsen

```
private void Metodnamn(object sender, EventArgs e)
{
    // Här skrivs de satser som tar hand om händelsen
}
```

Copyright 2015 - Mahmud Al Hakim www.webbacademy.se

Några händelser (events)

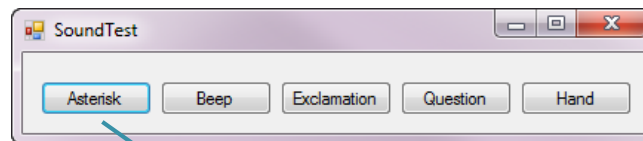
Click	klicka på komponenten
DoubleClick	dubbelklicka
Enter	komponenten blir vald
Leave	komponenten förlorar fokus
MouseEnter	muspekaren flyttas till komponenten
MouseLeave	muspekaren flyttas från komponenten

Fullständig lista på alla events finns här:
<http://goo.gl/E207Wp>

Copyright 2015 - Mahmud Al Hakim www.webbacademy.se

Events – Exempel I

- Programmet SoundTest (Click) sidan 322



```
private void button1_Click(object sender, EventArgs e)
{
    SystemSounds.Asterisk.Play();
}
```

Copyright 2015 - Mahmud Al Hakim www.webbacademy.se

Events – Exempel 2

- Programmet SoundTest (MouseEnter och MouseLeave) sidan 323

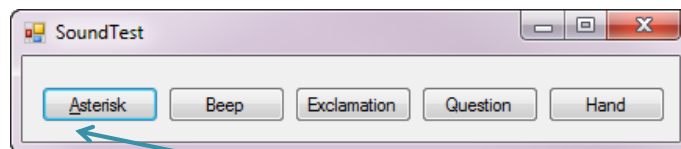
```
private void button_MouseEnter(object sender, EventArgs e)
{
    ((Button)sender).BackColor = Color.Red;
}

private void button_MouseLeave(object sender, EventArgs e)
{
    ((Button)sender).UseVisualStyleBackColor = true;
}
```

Copyright 2015 - Mahmud Al Hakim www.webbacademy.se

UseMnemonic (snabbtagenter)

- Med hjälp av egenskapen UseMnemonic (om satt till true) kan man använda **snabbtagenter**.
- Alt + en valfri tangent aktiverar en händelse.
- Tangenten bestäms genom att lägga in tecknet & i knappens text.



Text **&Asterisk**

Copyright 2015 - Mahmud Al Hakim www.webbacademy.se